Schema-Constrained Image Captioning for Five Low-Resource Indigenous Languages

Diego Cuadros[1], **Nicholas Leeds**[1], **Amanda Avalos**[1],
Azul Alipzar-Velazquez[1], Jared Coleman[1]
Faezeh Dehghan Tarzjani[2], **Bhaskar Krishnamachari**[2]
[1]Loyola Marymount University, [2]University of Southern California
dcuadros@lion.lmu.edu, nleeds@lion.lmu.edu, aavalo12@lion.lmu.edu,
aalpiza1@lion.lmu.edu, jared.coleman@lmu.edu
dehghant@usc.edu, bkrishna@usc.edu

Abstract

We describe our submission to all five tracks of the AmericasNLP 2026 Shared Task on Cultural Image Captioning: Bribri, Guaraní, Yucatec Maya, Orizaba Nahuatl, and Wixárika. Our system is an LLM-assisted rule-based machine translation (LLM-RBMT) captioner. For each language, a coding agent reads the small development split and open-web linguistic references and writes a complete Pydantic grammar package with a closed vocabulary. At inference time, a vision–language model sees the image and the schema, emits a structured `SentenceList` under constrained decoding, and a deterministic Python renderer produces the surface string. The model never generates target-language tokens. The same architecture handles all five languages with no fine-tuning, no parallel corpora, and no human edits to the generated packages. On the official test set, the system placed first on human evaluation in Bribri and Orizaba Nahuatl, third on Yucatec Maya, and first on ChrF++ in Yucatec Maya. We suggest that a strength of the approach is that outputs are restricted to simple sentences that are grammatically correct by construction, modulo the correctness of the generated grammar itself.

1 Introduction

The AmericasNLP 2026 Shared Task on Cultural Image Captioning (Bui et al., 2026) asks systems to produce target-language captions for culturally grounded images in five Indigenous languages of the Americas: Bribri (`bzd`), Guaraní (`grn`), Yucatec Maya (`yua`), Orizaba Nahuatl (`nlv`), and Wixárika (`hch`). The setting is intentionally adversarial for end-to-end neural captioning: each language includes only fifty development examples, no parallel image–caption corpora exist for training, and public linguistic references are uneven in quality and coverage. Submissions are first ranked by ChrF++, and a top-five-per-language subset is forwarded to human evaluation by speakers.

A direct neural approach to this task asks a vision–language model (VLM) to caption the image in the target language. This works in proportion to how much of each target language the model has seen in pretraining, which for these five languages ranges from almost nothing to occasional Bible translations and Wikipedia stubs. The model can still produce fluent-looking strings, but it has no way to tell the user that it is guessing, and the user has no way to tell which fragments of the output are grounded.

We take a different approach. Rather than trying to make a VLM better at decoding into a language it does not know, we remove its ability to decode into the target language at all. For each language we generate a *language package*: a small Python module that encodes a closed vocabulary, a set of sentence templates, and a deterministic renderer. The VLM is then constrained to emit a structured object that conforms to the package's Pydantic schema, and the rendered string is produced by Python. Because the schema's lexical fields are typed as `Literal[...]` enums drawn from the package's vocabulary, the VLM cannot emit an out-of-vocabulary lemma; the constraint is enforced during decoding by the structured-outputs API. A narrow `proper_noun` field is the single escape hatch. The full source code for the captioner, the generator agent, and the five generated language packages is publicly available.[1]

Because none of the authors speak any of the target languages, the language packages themselves are also not written by us. Instead, they are written by a coding agent that reads the development split, web search, and reference packages, then writes a self-contained Python module that imports, validates, and renders against held-out captions. The agent is required to cite each external linguistic

[1] https://github.com/kubishi/americasnlp-2026-shared-task

Proceedings of the Sixth Workshop on NLP for Indigenous Languages of the Americas (AmericasNLP), pages 257–263

reference inline in the package's own documentation, so that any claim made by the grammar can be traced back to the source the agent learned it from. The agent is the per-language adaptation cost; once it has produced a package, captioning is a single VLM call per image.

The resulting packages are *auditable*: a reviewer can read the vocabulary, sentence templates, and renderer source, and trace any surface caption back through the structured intermediate to the specific lexical and grammatical choices that produced it, each of which is cited in the package's inline documentation. Auditability is not the same as correctness, however. A grammar that the agent generated from public references and a fifty-row development split, with no speakers in the loop, can be fully traceable and still wrong. Any practical use of this architecture requires that the generated packages be validated by native speakers and community partners.

Contributions. This paper makes the following contributions.

1. A schema-constrained image-captioning architecture for low-resource Indigenous languages in which the VLM never generates target-language tokens. Lexical fields are typed as strict `Literal[...]` enums drawn from a closed vocabulary, a deterministic Python renderer produces the surface string, and a single narrowly prompted `proper_noun` slot carries genuine named entities through verbatim.
2. A reproducible agent-driven workflow for writing language packages from the shared task's development split and open-web references, applied uniformly to five typologically distinct languages with no per-language tuning by the authors.
3. Development-set comparisons against pipeline and direct-prompting baselines, together with the official test results, and an analysis of the dissociation between ChrF++ and human-evaluation ranks under which the system placed first on human evaluation in two of five languages and top-three in a third.

Outline. Section 2 situates the work relative to LLM-RBMT, low-resource MT, and structured decoding. Section 3 describes the language packages and the captioning pipeline. Section 4 states the experimental setup. Section 5 reports development and final test results. Section 6 discusses error patterns and the gap between ChrF++ and human evaluation. Section 7 states the system's limitations and Section 9 concludes.

2 Related Work

LLM-assisted rule-based MT. Our system is a vision-input extension of the LLM-Assisted Rule-Based Machine Translation (LLM-RBMT) paradigm introduced for endangered-language text-to-text translation (Coleman et al., 2024, 2026a). The original LLM-RBMT systems used an LLM to decompose English input into a structured intermediate that an expert-designed renderer mapped into the target language. We adopt the this approach, replacing the text encoder with a VLM, and the per-language hand engineering with an agent that writes the renderer.

Low-resource neural MT. A large literature evaluates massively multilingual neural MT models (e.g., NLLB-style systems) on low-resource languages (NLLB Team et al., 2024). These systems perform well when the target language has at least modest parallel data and degrade sharply otherwise. For the AmericasNLP languages, parallel data is either tiny or absent, and the organizer baseline (Qwen3-VL $\rightarrow$ NLLB) reflects this: it covers four of five target languages and reaches a four-language mean of 14.4 ChrF++.

Structured decoding and constrained generation. Recent work uses constrained decoding, structured outputs, and JSON schemas to force LLMs to emit valid objects (OpenAI, 2024). Our contribution is not the constraint mechanism itself but its use as the *only* interface to the target language. The schema is generated per language and encodes the grammar. The LLM is structurally unable to bypass it.

Agent-authored code and grammars. Coding agents have recently become capable enough to write self-contained Python modules that pass tests. We use this capability to push per-language engineering effort into the agent rather than the authors. Earlier LLM-RBMT systems required substantial human time per language (Coleman et al., 2026a). Ours requires only that the agent's run succeed.

3 System

The system has two parts. A *generator* runs once per language and emits a language package. A *cap-*

```
yaduha-{iso}/
  pyproject.toml
  yaduha_{iso}/
    __init__.py
    vocab.py
    prompts.py
```

Figure 1: Language package layout. `vocab.py` lists the closed vocabulary; `__init__.py` defines the Pydantic `Sentence` subclasses and the deterministic `__str__()` renderers; `prompts.py` carries any language-specific guidance the agent wanted the VLM to see.

tioner runs once per image and uses the package as its schema. Once a package exists, the captioner is a single VLM call followed by a deterministic render. Our packages follow the YADUHA[2] convention used by prior LLM-RBMT work for endangered languages (Coleman et al., 2024, 2026a).

3.1 Language Packages

Each language package is a standard Python package with the layout shown in Figure 1.

The package exports a `language` object with a small number of sentence classes (typically subject–verb, subject–verb–object, and a copular or stative variant where the language has one). Each class is a Pydantic model with morphological fields (person, number, tense or aspect, possession, and so on) and lexical fields whose types are `Literal[...]` enums drawn from `vocab.py`. The class's `__str__()` method renders the structured object into a surface string in the target language. Rendering is pure Python; no LLM runs at render time.

Strict-literal lemma typing. One important schema decision is that lexical fields are strict enums. If `vocab.py` lists thirty nouns, the noun-lemma field is typed `Literal["noun_1", ..., "noun_30"]`. The VLM's structured-outputs decoder refuses to emit any other string. The effect is that out-of-vocabulary nouns cannot leak into the output disguised as fluent target-language words.

Proper-noun escape hatch. Strict vocabularies are appropriate for common nouns. They are too strict for the named entities that cultural captions routinely contain (places, monuments, markets, civic institutions). Every `Noun` model therefore exposes a single `proper_noun: Optional[str]` field. The VLM may write a name into this field when the image contains one; otherwise it is left empty. The system prompt instructs the model not to abuse the escape hatch as a way around the strict lemma constraint.

3.2 Captioning Pipeline

The submitted pipeline is one VLM call per image:

$$\text{image} \rightarrow \text{VLM} + \texttt{schema} \rightarrow \texttt{SentenceList} \rightarrow \texttt{render()} \rightarrow \text{caption}.$$

The VLM (`gpt-5`) receives the image and the Pydantic schema generated from the language package and emits a `SentenceList` JSON object through the OpenAI structured-outputs API. Python then renders each `Sentence` in the list and concatenates the results.

3.3 Generator Agent

For each target language, the generator agent receives a small training slice of the available captions, the URLs of public linguistic references, one or more existing Yaduha packages as templates, and a validation harness. The harness exposes three tools: a package importer and schema-sanity checker, an English-to-structure smoke test, and a held-out comparison against a disjoint validation slice of captions. The agent reads, writes, and tests Python code in a loop until the harness reports a valid, self-consistent package.

The agent's bootstrap prompt forbids hardcoded shortcuts of the form "if the English input is exactly X, output Y." Every behavior must route through structured `Sentence` inputs and the rendering logic. The held-out validation captions are visible to the harness for scoring but not to the agent, and the harness reports only aggregate statistics. Absent native-speaker review, this is the best automatic check we have that the agent is generalizing rather than memorizing.

We implemented the generator using Anthropic's Claude Opus 4.7 model invoked through the Claude Code coding-agent toolkit, which exposes the file-system and shell tools needed to read references, write Python modules, and run the validation harness in a loop. The architecture does not depend on this specific agent.

4 Experimental Setup

Languages and data. We use the official AmericasNLP 2026 splits without modification. Table 1 summarizes sizes. For the generator, we partition

[2] https://github.com/kubishi/yaduha

Language	ISO	Dev	Test
Bribri	bzd	50	267
Guaraní	grn	50	110
Yucatec Maya	yua	50	212
Orizaba Nahuatl	nlv	50	200
Wixárika	hch	50	201
Total		250	990

Table 1: Dataset sizes. The agent sees only a deterministic 30-row slice of each development split. The remaining 20 rows are held out for validation.

each fifty-row development split into a deterministic thirty-row training slice (exposed to the agent as input captions) and a twenty-row validation slice that is hidden from the agent and used by the harness for held-out scoring. All development numbers reported below are computed over the full fifty-row split. Test scores come from the official test set.

Metric. Official ranking uses ChrF++ (Popović, 2015) in the first stage and human evaluation by speakers (top five systems per language) in the second stage. Per-row ChrF++ in our development tables is computed with `sacrebleu`'s default settings. The mean over a language is the unweighted mean of per-row scores. The test ChrF++ in Table 4 is the official score reported by the organizers.

Configurations. We evaluate the configurations listed in Table 2. The submitted configuration is the one-step `gpt-5` captioner with the minimal v3 prompt. The two pipeline configurations differ only in their structured-translator backend. The direct three-shot baseline asks a strong VLM (`claude-sonnet-4-5`) to produce a target-language caption given three in-context examples. The organizer baseline is the shared task's official Qwen3-VL → NLLB system. We report its published per-language ChrF++ where available.

5 Results

5.1 Development Set

Table 3 reports per-language mean ChrF++ on the fifty-row development split for every evaluated configuration. The final column is the unweighted mean over the four languages on which the organizer baseline reports a number. This is the column on which our system can be compared directly with the official baseline.

Three observations follow. First, the schema-constrained captioner (which never produces a target-language token by free generation) beats the direct-prompting baseline by 0.84 ChrF++ on average and beats the organizer baseline by 3.30 ChrF++ on the four-language mean. The improvement is largest on Yucatec Maya (uncovered by the baseline) and Orizaba Nahuatl. Second, the schema-constrained captioner trails the organizer baseline on Guaraní and Wixárika, both languages where NLLB has some training data and the package coverage is comparatively thin. Third, within our family, the simplest method wins: one-step schema-constrained captioning outperforms every two-step pipeline variant on average, including the variant that uses `gpt-5` as the translator.

Configuration	Description
`gpt-5` one-step v3 (submission)	Image + Pydantic schema → `SentenceList` → Python render. Minimal prompt.
`gpt-5` one-step v2	Same as above with the earlier, more verbose prompt.
Pipeline (sonnet + 4o)	VLM English caption → `gpt-4o` structured translator → render.
Pipeline (sonnet + 4o-mini)	As above with `gpt-4o-mini` translator.
Pipeline (sonnet + gpt-5)	As above with `gpt-5` translator.
Direct 3-shot	`claude-sonnet-4-5` caption in target language given three in-context examples.
Organizer baseline	Qwen3-VL → NLLB (no coverage for `yua`).

Table 2: Evaluated configurations.

5.2 Final Test Results

Table 4 reports the official results released by the shared-task organizers. ChrF++ ranking is computed over all submitted systems. Human evaluation considers the top-five-per-language ChrF++-ranked systems, with mean rating from speaker evaluators on a 4-point scale.

The system placed first on human evaluation in Bribri and Orizaba Nahuatl, third in Yucatec Maya, and did not qualify for human evaluation in Guaraní or Wixárika. ChrF++ ranks are mid-pack in three languages and last-but-one in two. The dissociation between ChrF++ and human rank is the central empirical finding of this paper and we discuss it in Section 6.

6 Analysis

6.1 ChrF++ Versus Human Judgment

The system's results split cleanly along the ChrF++/human-rating axis. On the three languages

Configuration	bzd	grn	yua	nlv	hch	Mean†
`gpt-5` one-step v3 (submission)	**11.93**	17.47	**27.71**	**26.64**	14.84	**17.72**
`gpt-5` one-step v2	11.73	19.17	26.02	24.79	16.20	17.97
Pipeline (sonnet + gpt-5)	10.84	15.60	17.44	17.27	16.16	14.97
Pipeline (sonnet + 4o)	10.96	15.63	17.02	16.79	14.73	14.53
Pipeline (sonnet + 4o-mini)	10.38	15.32	16.48	16.81	15.56	14.52
Direct 3-shot (sonnet 4.5)	9.43	18.37	18.86	21.56	**18.14**	16.88
Organizer baseline (Qwen3-VL → NLLB)	7.57	**20.82**	—	11.53	17.77	14.42

Table 3: Development-set per-language mean ChrF++ (N=50 per language). **Bold** marks the best score per column. †Mean is over the four organizer-comparable languages (`bzd`, `grn`, `nlv`, `hch`); `yua` is excluded because the organizer baseline does not cover it.

Language	RAN	ChrF++ score	ChrF++ rank	Human rating	Human rank
Bribri	3 / 2 / es-3	10.03	5 / 7	2.895	**1 / 5**
Yucatec Maya	5 / 4 / es-4 / en-4	**23.41**	**1 / 6**	2.892	3 / 5
Orizaba Nahuatl	6 / 4 / es-4 / en-3	21.00	2 / 7	3.465	**1 / 5**
Guaraní	6 / 1 / en-6 / es-2	16.90	7 / 8	—	DNQ
Wixárika	4 / 2 / es-4 / en-4	15.61	7 / 8	—	DNQ

Table 4: Official test results for team `yaduha`. RAN (Resource Abundance Notation) (Coleman et al., 2026b) reports the order-of-magnitude count of speakers / monolingual / bilingual partners for each language. **Bold** marks first-place finishes. "DNQ" indicates that the system's ChrF++ rank did not place it in the top-5-per-language subset that was forwarded to human evaluation.

where it qualified for human evaluation, it took two firsts and a third. On the two where it did not qualify, its ChrF++ rank was near the bottom of the field. This pattern is consistent with the architecture's design intent: outputs are grammatical by construction (insofar as the package's grammar is correct), but the package's vocabulary is bounded, so character overlap with diverse human references is capped.

ChrF++ rewards character-level overlap with a single reference caption. A system that uses a closed vocabulary and a small set of sentence templates can be near-perfect on the constructions it covers, but it incurs a fixed cost whenever the reference uses a word or construction the package lacks. A system that fluently generates target-language strings, by contrast, will often score higher on ChrF++ even when the strings are grammatically or semantically broken, because the references they are scored against are themselves character n-grams. Human evaluators see through this in a way the metric does not. On the other hand, because the schema admits only a small set of sentence templates and a closed vocabulary, the captions the system produces are necessarily shorter and more uniform than what a free-form generator might write. The stylistic range a fluent speaker would draw on is unavailable to the system, and this is visible to human evaluators as well.

6.2 Where Coverage Hurts

The two languages on which the system failed to qualify, Guaraní and Wixárika, illustrate the cost of closed vocabularies, though we are limited in what we can say about why. None of the authors are linguists or speakers of any of the target languages, so the explanations below are not first-hand diagnoses but a post-hoc analysis produced by the same coding agent that wrote the packages, asked to read the failing predictions against the references and conjecture about the gap. For Guaraní, the agent attributes the failure to productive morphology (possessive prefixes and person-marked verbs) that its package modeled only partially, leaving surface strings as reasonable base forms but missing the inflectional patterns the references use. For Wixárika, the agent points to a richer property-predication system than its package captured. After we removed a degenerate `CopularSentence` type that had been producing tautologies of the form "X is X," the package lost its only path to express adjective-like content. If these explanations are correct, both gaps would respond to better packages rather than to larger models, but verifying them and identifying the gaps we are missing would require a post-mortem with linguists and speakers of the

affected languages, which we view as a valuable direction for follow-up work.

6.3 Where Constraint Helps

The Bribri result is the most striking case in this direction. The system's ChrF++ score placed it 5th of 7 submissions (just making the top-five-per-language cutoff for human evaluation) and yet speaker evaluation ranked it first. This is consistent with the picture sketched above: the captions were not the most fluent in the field, but the constrained-generation mechanism kept them grammatical and prevented the model from producing fluent-looking but ungrammatical/incorrect strings that the evaluators would have rejected, and that appears to have weighed more heavily in speaker judgment than character overlap weighed in the metric. Orizaba Nahuatl tells a milder version of the same story (2nd on ChrF++, 1st on human evaluation). The natural counterfactual question concerns Guaraní and Wixárika, where ChrF++ ranks of 7th out of 8 placed the system below the cutoff for human evaluation. We cannot say how those captions would have fared under speaker review, but the Bribri result is at least suggestive that the bottom of the ChrF++ table is not necessarily a reliable predictor of the bottom of the human-evaluation table.

6.4 One-Step Versus Two-Step

In the pipeline configurations, a VLM first produces a free-form English caption of the image, and a text model then maps that English caption into the structured `SentenceList` that the renderer consumes. The problem with this approach is that information is lost at each handoff: the VLM commits to an English description before knowing which of the package's lemmas and sentence types are available to express it, and the downstream translator has to recover the structured intermediate from text that may no longer carry the necessary cues. The one-step approach avoids this by letting the VLM select the structured fields directly, conditioned on the image. Table 3 shows that on every language except Wixárika, the one-step pipeline beats every two-step variant, including the variant that uses the same `gpt-5` model as the structured translator.

7 Limitations

We note four limitations of the system.

Package coverage is a hard ceiling. A schema-constrained captioner cannot produce a concept the package does not express. For languages where the package's vocabulary or morphology is thin, the system can produce a grammatical and fully auditable output that nevertheless misses what the reference caption says. The fix is better packages, which ultimately requires more time with the agent or (preferrably) with speakers.

ChrF++ underestimates the architecture. The dissociation between ChrF++ rank and human rank in Table 4 is suggestive, not conclusive: it could reflect a general property of schema-constrained systems, or it could reflect specifics of this particular shared task. Either way, the system is at a structural disadvantage on the metric used for the first ranking stage.

No native-speaker review of packages. We did not have native-speaker validation of the agent-generated packages during the shared-task window. The official human evaluation is the strongest external check we have, and it is positive on three of five languages, but a serious deployment of this architecture would require collaboration with speakers.

Frontier-model dependence. The submitted configuration uses `gpt-5` as the VLM. We evaluated open-weight VLMs (`qwen2.5-vl:32b`) during development and found that they satisfy strict schemas less reliably without fine-tuning, especially when vocabularies grow large.

8 Ethics Statement

This work concerns Indigenous languages and culturally grounded images. The system is designed so that errors surface in the output rather than being hidden: when the package cannot express a concept, the constrained-output mechanism leaves a visible gap rather than producing a fluent-looking guess. Aditability is not the same as correctness, though, and we did not have native-speaker validation of the agent-generated packages during the shared task. The captions produced by this system should be treated as experimental.

Any deployment of this architecture beyond an development setting should involve close collaborations with speakers and community members. The agent's web search consumes public linguistic references whose authorship includes both community-authored materials and secondary

summaries. Those resources should be cited and weighted accordingly in any published package.

9 Conclusion

We presented a schema-constrained image-captioning system for five low-resource Indigenous languages in which a vision–language model never generates target-language tokens. Surface strings are rendered deterministically from a structured intermediate whose lexical fields are typed as strict enums drawn from an agent-authored language package. The same architecture handles all five languages with no fine-tuning, no parallel data, and no human edits to the packages.

On the official shared task, the system placed first on human evaluation in Bribri and Orizaba Nahuatl, third in Yucatec Maya, and first on ChrF++ in Yucatec Maya. It did not qualify for human evaluation in Guaraní or Wixárika, where ChrF++ ranks were weak. We read this dissociation as evidence that schema-constrained systems are penalized by character-overlap metrics and rewarded by speaker evaluation, and we think the architecture's value is most visible in the latter.

References

Minh Duc Bui, David Guzmán, Abteen Ebrahimi, Franklin Morales, Marvin Agüero-Torales, Raquel Insfrán, Cecilia González, Ramón Araujo, Luca Cernuzzi, Carlos Raul Noh Chi, Carlos Eduardo Tec Cahun, Sindi Estrella Poot Cohuo, Daniel Ricardo Benítez Chi, Santos Natanael Palomo Arévalo, Jessica Elizabeth Canul Canche, Deysi Aracely Poot Poot, Wendy Marleny Dzib Dzib, Eduardo José Ake Pool, Reynaldo Alexander Couoh Martin, and 15 others. 2026. Findings of the AmericasNLP 2026 shared task on image captioning in Indigenous languages. In *Proceedings of the Sixth Workshop on NLP for Indigenous Languages of the Americas (AmericasNLP)*, San Diego, California. Association for Computational Linguistics.

Jared Coleman, Bhaskar Krishnamachari, Ruben Rosales, and Khalil Iskarous. 2024. LLM-assisted rule based machine translation for low/no-resource languages. In *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP 2024)*, pages 67–87, Mexico City, Mexico. Association for Computational Linguistics.

Jared Coleman, Ruben Rosales, Kira Toal, Diego Cuadros, Nicholas Leeds, Bhaskar Krishnamachari, and Khalil Iskarous. 2026a. Comparing LLM-based translation approaches for extremely low-resource languages. In *Proceedings of the Ninth Workshop on Technologies for Machine Translation of Low Resource Languages (LoResMT)*. Association for Computational Linguistics.

Jared R. Coleman, Tainã G.D. Coleman, and Bhaskar Krishnamachari. 2026b. RAN: Resource abundance notation for languages in NLP. In *Proceedings of the Sixth Workshop on NLP for Indigenous Languages of the Americas (AmericasNLP)*, San Diego, California. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2024. Scaling neural machine translation to 200 languages. *Nature*, 630(8018):841–846.

OpenAI. 2024. Introducing Structured Outputs in the API. https://openai.com/index/introducing-structured-outputs-in-the-api/.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.